· <u>REMARKS</u>

Applicant wishes to first review what is discussed in the background of the invention. The software development for applications on digital signal processors is an iterative process. A compiler translates source code into an assembly language source code and an assembler translates the assembly language source code files into machine language object files. A linker combines the object files into a single executable object module or program. The linker accepts object files created by the assembler and accepts archive or library members and output modules created previously. The objective of this development process is to produce an executable module that may be stored and executed on a device or processor. A linker combines the object files into a single executable object module or program.

Debugging tools are available to test processors and the linked, executable code. Tools for debugging software in a system context include simulators and emulators. An emulator is a software development tool that allows software under development to be executed, controlled, and viewed in a real hardware environment. An emulator allows a user to perform software and hardware development, and to integrate the software and hardware with the target processor. The most desirable form of emulator allows software development to occur in the real product hardware environment. To allow this form of emulation, the target processor provides an emulation interface that accesses its internal state. An emulation interface provides control and access to every memory location and register of the target processor, and extends the target processor architecture as an attached processor. The emulator can start or stop execution of the target processor program. When the target is stopped, the emulator has the capability to load, inspect, and modify all processor registers. Program data and program memory may be upgraded or downloaded.

Increasingly, embedded systems are being built the purpose of which in whole or in part involves communication. In many of these cases the product design involves more than one processor. Frequently, one processor is a micro-controller and another is a digital signal processor.

In a multiprocessor system, there is one set of development tools (compiler, assembler, linker, perhaps debugger) for each kind of processor. Traditionally,

there was also an emulator for each kind of processor. However, as integrated circuit technology makes it increasingly practical to put the entire system on a chip, it becomes necessary to provide debug connection to multiple processors via a single hardware debug port. With traditional emulation technologies, there were three alternatives available:

1. Use a different emulator for each kind of processor. This would necessitate connecting/disconnecting emulators to switch between processor views. Only one processor would be debugable at a time.

2. Use a single emulator with target interface software that understood a particular kind of processor. To switch to a different processor view, different target interface software must be used. Only one processor would be debug-able at a time.

3. Use a single emulator with target software built to understand the particular combination of processors in the system. In this scheme, all processors are debug-able at once. However, to debug a different kind of target system (with different processor kinds), new emulation software must be built.

The first two schemes are inconvenient for the user who would like to be able to debug all processors on his system at the same time. The third scheme is difficult for the emulator provider, who must produce new target interface software every time an application with a different combination of processors arises.

From the foregoing it may be appreciated that a need has arisen to provide emulation capability that can be used with a variety of target systems involving different mixes of target processors.

Claim 1 is rejected under 35 U.S.C. 102 (e) as being anticipated by Proteat et al. (U.S. Patent no. 5,970,245; hereinafter Proteat).

The present application uses dynamic linking to facilitate the distribution of DLLs between a host and emulation system to facilitate the operation of a multiprocessor capable In-Circuit emulator. Applicant's Claim 1 calls for "In a data exchange system for transferring data between a host processor and a target processor comprising:

a data unit on said target processor that transfers said data from said target processor to an emulator and a device driver on said host processor that transfers data from said emulator,

a system for dynamically linking and loading software support for a new target processor comprising:

a target interface module for the host computer that supports the new target processor kind; and

a target interface module for the emulator that supports the new target processor kind."

The examiner argues that applicants' claims are made by Proteat. This patent describes a method for tracing a DLL's (Dynamic Linked Libraries) API calls within a single processor system. Referring to Fig. 1 it shows a single computer with a single processor 12. The description of Fig. 1 is that it is a block diagram that illustrates an exemplary hardware environment of the present invention. The memory includes an operating system, software applications and a trace DLL and a target DLL. There is only one processor. The examiner references col. 2, lines 36-47 but nowhere does it teach or discuss more than one processor. There is no teaching of a data exchange system for transferring data between a host processor and a target processor. There is no a data unit on said target processor that transfers said data from said target processor to an emulator and a device driver on said host processor that transfers data from said emulator. The examiner also references Col. 3, lines 13-23 but there is no teaching of more than one processor or of a data exchange system for transferring data between a host processor and a target processor. There is a target procedure in a target DLL but no target processor or loading emulation software support for a new target processor. Col. 2, lines 48-64 do not mention multiprocessor including a new target processor. This patent describes a method

7

for tracing a DLL's (Dynamic Linked Libraries) API calls within a single processor

system.

Claim 2 calls for " A method for at time of use linking and loading of emulation software for one or more debuggers on a host computer to communicate with a mix of target processors via a JTAG debug link and emulator comprising the steps of: connecting a debugger for each processor to a target interface for that kind of processor; determining if there is support for that kind of processor in the emulator by the target interface communicating with a dynamic loader on the host computer; if not support loading a target interface into the emulator and connecting to an already running emulation software on the host computer; and connecting the target interface software on the emulator to the target interface software on the host computer."

The specification of the reference patent of Proteat has been reviewed and this patent describes a method for tracing a DLL's (Dynamic Linked Libraries) API calls within a single processor system. The patent does not teach linking and loading of emulation software for one or more debuggers on a host computer to communicate with a mix of target processors via a JTAG debug link and emulator. There is no step in the reference of connecting a debugger for each processor to a target interface for that kind of processor. There is no determining if there is support for that kind of processor in the emulator by the target interface communicating with a dynamic loader on the host computer and no step of if not support loading a target interface into the emulator and connecting to an already running emulation software on the host computer; and connecting the target interface software on the emulator to the target interface software on the host computer. The examiner references Col. 4, lines 21-63 but that text only describes a method for tracing DLL API calls within a single processor. JTAG interconnect is known in and described in IEEE standard 1149.1.
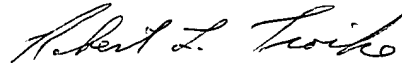
Claim 3 calls for the method of Claim 2 wherein said steps are repeated for each debugger, for each kind of processor on the target system. Claim 3 is therefore deemed allowable for at least the same reasons as Claim 2. It also calls for repeating the steps for each kind of processor. There is only a single processor in the Proteat reference.

Claim 4 calls for "The method of Claim 2 wherein said determining step includes communicating using ECOM modules on the host computer and emulator over a host computer to emulator connection." Claim 4 dependent on Claim 2 is deemed allowable for at least the same reasons as Claim 2. The claim further describes using ECOM modules on the host computer and emulator in a host computer to emulator connection. No such connection is suggested in the Proteat reference. ECOM is well known term of art that refers to common embedded component module as shown and described in Fig. 1.

Claim 5-8 are deemed allowable over Proteat for the reasons presented above.

In view of the above applicants' claims 1-8 are deemed allowable and an early notice of allowance of these claims is deemed in order and is respectfully requested.

Respectfully submitted;

Robert L. Troike (Reg. 24183)

Tel. No. 301-259-2089